Investigations in the Use of Clinical Algorithms to Organize Medical Knowledge

Thomas W. Abendroth, MD
Robert A. Greenes, MD, PhD
Elizabeth A. Joyce, MBBS

Harvard Medical School, Decision Systems Group
Brigham and Women's Hospital, Department of Radiology
Harvard School of Public Health, Department of Biostatistics

Abstract

Clinical algorithms can concisely portray the often complex branching structure of diagnostic and therapeutic strategies. Though frequently used in the literature to present expert opinion regarding specific problems, algorithms suffer a number of limitations when restricted to a static medium. We have investigated ways in which computer presentation might overcome these limitations, and developed techniques for handling some of the challenges encountered in adapting printed algorithms to computer display. Based in part on this work, we believe clinical algorithms could play a useful role in decision support and context-sensitive knowledge retrieval.

Introduction

Algorithms (flow charts) are a common means of presenting expert judgment concerning diagnostic and therapeutic strategies. They concisely depict the logic of clinical decisions, explicitly identifying the pertinent elements in the decision process. Their branching structure captures the relationships among these elements with a clarity difficult to achieve using linear text.

Formal algorithms, however, are used sparingly in medical practice. While they have allowed physician assistants and nurse practicioners to extend their roles as health care providers [1,2], physician use and acceptance has been especially limited [3] [see 4, 5 for notable exceptions]. Incompleteness and rigidity are commonly cited as major deficiencies of this approach. Yet, the popularity of algorithms in the clinical literature [6-11] provides evidence for their usefulness in organizing knowledge and presenting expert opinion. This role might be further expanded with computer implementation.

Advantages of computer implementation

To imagine the potential uses of clinical algorithms, one must look beyond the algorithm as a rigid step-by-step procedure for solving a problem. That restricted view has inspired much criticism of the printed algorithm as a decision support tool. With a more flexible presentation, clinical algorithms might provide both a framework for decision making and a starting point for context-sensitive information retrieval.

The static nature of printed algorithms effectively limits their appeal in decision support, for two major reasons. First, the complexity of a printed algorithm is restricted to what can be displayed on a handful of pages. If this size is exceeded, the algorithm becomes difficult to follow. Second, the printed algorithm is usually shown as a single "flat" view, displaying all possible paths in their entirety. Even when the user is interested in only one particular portion of the algorithm, the details of all other portions remain visible, cluttering the page.

With computer presentation, the algorithm is not limited by these static constraints. The display can change to reflect information and choices supplied by the user. Paths which already have been excluded can be pruned from view, and the user can control the level of detail to be shown. A variety of browsing, abstracting and zooming techniques enable computer presentation of large algorithms to be more manageable than is possible in a printed format. With this flexibility, the complexity of the algorithm is no longer limited to a size conveniently displayed on a few printed pages. One might even imagine displaying extremely complex "algorithms" generated dynamically using an underlying rule base and inference engine.

Algorithms as a starting point in knowledge retrieval

Clinical algorithms cannot be absolute or all-inclusive. They can, however, provide a useful starting point in the decision making process. As a starting point, each algorithm should not only suggest a sequence of actions for handling a specific problem, but also guide the interested user to background and related knowledge.

We have adopted the convention that each step in the algorithm is associated with a readily accessible narrative discussion which further explains the rationale and/or details for that step. From this narrative discussion, links can be provided to other pertinent issues, alternative approaches, evidence justifying the particular strategy proposed, related material, etc. We anticipate that links to relevant images (eye grounds in hypertensive retinopathy, EKG showing left ventricular strain) or audio segments (auscultation of third and fourth heart sounds) will also be helpful. For the physician who is using an algorithm primarily to facilitate retrieval of information relevant to a specific problem, the flow chart diagram itself may be less important than the supporting documentation to which it provides quick access.

Algorithms in decision support

A library of clinical algorithms might eventually become one component of the physician's workstation, a computer environment which provides a wide range of decision support tools. Other tools would include online medical references, medical records, and diagnostic aids (such as QMR [12] or DXplain [13]). Given the volume of information literally at the physician's fingertips, the workstation will need to support quick access to the small fraction of material pertinent to any particular problem. A number of tools could help in this regard: a hypertext tool linking related information across textbook or journal article boundaries, a tool for perusing MeSH listings and constructing appropriate literature searches, a tool for cataloging and retrieving images, and many others.

Clinical algorithms might also be helpful in this process.

Ideally, the physician would be informed automatically whenever a patient's status triggered one of the protocols in an algorithm library. By consulting a particular algorithm, the physician could speed information access in at least two ways. First, as mentioned earlier, the algorithm would contain not only a suggested approach to the problem of interest, but also links to specific pertinent background and related material. Second, and potentially more important, the clinical algorithm would define the context in which decision support is needed. This defined context could help the system anticipate the kinds of questions the physician might have, and the kinds of information the physician might desire. Preliminary work suggests that algorithms may also provide an intuitive interface for patient data entry [14].

Even without the support of the physician's workstation, however, the clinical algorithm provides a useful framework for structuring related information. It defines a world of restricted breadth and depth, within which to examine a specific problem. By limiting the number of questions/decisions considered potentially relevant, the algorithm focuses the physician's attention on issues which may be important to assess, and the reasons for their potential importance. In this mode alone, the algorithm provides a starting point for formulating a strategy.

The physician-computer interface

To use algorithms effectively in computer-based knowledge retrieval and decision support, one must have an acceptable interface for the physician-computer interaction. Creating this interface has been the focus of our current work. This task presents a number of interesting challenges, which stem from the fact that users must be able to: (1) browse through complex algorithms without becoming lost, and (2) vary the level of detail displayed as interest shifts from one portion of the algorithm to another.

As discussed above, the need to restrict a paper-based algorithm to a few pages effectively limits its appeal as an information management tool. While the computer provides more flexibility in presenting algorithms, limited display size is still an issue. Many computer screens are smaller than a single sheet of paper, and most are smaller than the three or four sheets of paper often required to display a modest algorithm in its entirety. For a computer implementation to overcome the restrictions which hamper the printed format, it must provide an effective means to keep the user oriented to his or her position within the algorithm.

We have explored a number of techniques to maintain user orientation. Most rely upon conventions to which the user becomes accustomed, so that new situtations are familiar in appearance and form, if not in detail. Let us divide the orientation problem into two broad categories: maintaining *local orientation*, and maintaining *global orientation*. Techniques for maintaining local orientation allow the user to quickly regain his or her bearings after momentarily looking away from the screen. Techniques for maintaining global orientation keep the user aware of his or her position within the algorithm as a whole.

Maintaining local orientation

A common technique for maintaining local orientation is to use different symbols to signify different types of actions within the algorithm [6, 9-11]. For example, each step in the algorithm might be displayed within a box which has one of three shapes:

- round cornered rectangles to designate clinical states (such as "Adult with no total cholesterol determination in past 5 years")
- square cornered rectangles to denote actions which must be performed (such as "Do non-fasting total cholesterol")
- hexagons to signify branch points where a result value (such as a test result: "Result of cholesterol determination?" or an assessment: "Are coronary artery disease or other risk factors present?") is required to determine the next appropriate action.

With this convention, the user can recognize the function of each step at a glance, even without knowing the details.

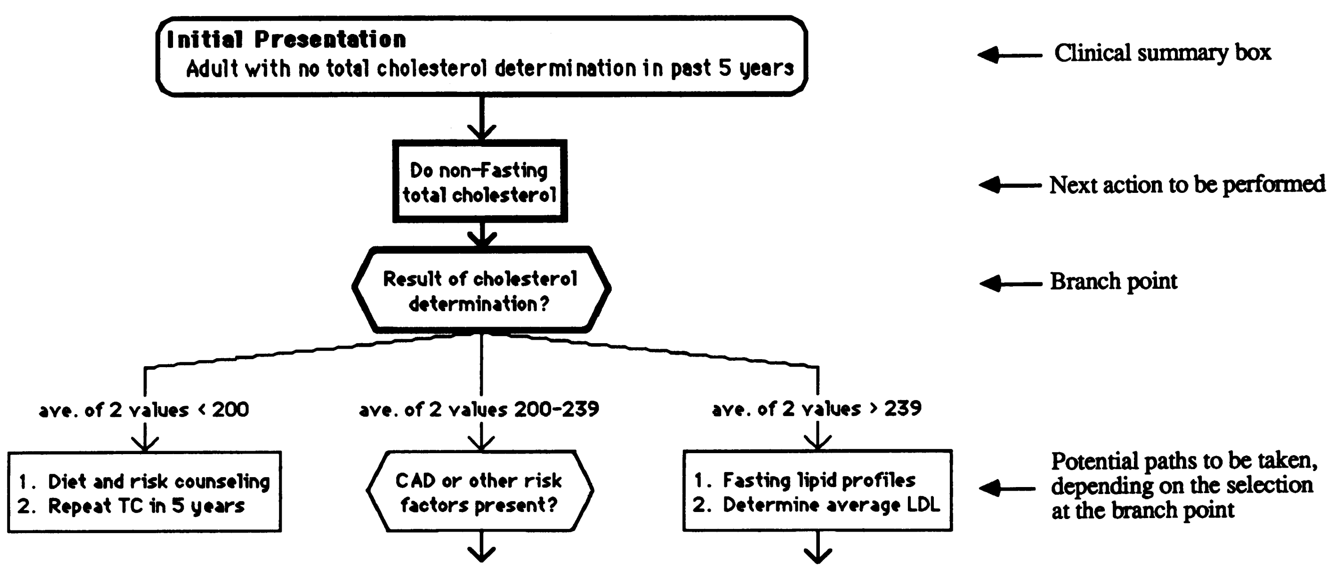

Figure 1. The left side of the figure shows the initial display for an algorithm concerning routine screening for hyperlipidemia. At this point, the clinical summary box contains only a statement of the clinical presentation. On the right side of the figure are labels (which are not part of the display) identifying the various portions of the display.

Maintaining global orientation using physical position

Let us now consider three general approaches to maintaining global orientation, which can be especially challenging when the view size is small and the algorithm complex. The first approach attempts to keep the user oriented to his or her *physical position* within the algorithm (for example: you got here by taking the leftmost path from the first step in the algorithm, then the middle path from the second step...). This approach is often implemented using a miniaturized view of the entire algorithm, with the current position highlighted in some fashion. This technique is particularly helpful when:

- the user is already familiar with the physical layout of the algorithm
- the algorithm is simple enough that the user can easily distinguish the different functional areas in the miniaturized view, and remember the function associated with each

These are especially stringent requirements, however, in that this technique is least helpful when help is needed most (an unfamiliar user and a complex algorithm). Moreover, this approach may unreasonable if the algorithm is generated dynamically using an underlying rule base. In this case, even frequent users may not feel comfortable with a miniaturized view of a complex structure which takes on many different forms (depending on the rules invoked).

Maintaining global orientation using logical position

A second approach to maintaining global orientation is to keep the user aware of his or her *logical position* in the algorithm, without reference to the physical position. The user's logical position is determined by the relevant information already obtained, regardless of the temporal sequence or particular pathway followed in collecting this information.

With our current implementation of this technique, each display includes a "clinical summary box," which records pertinent data already known about the patient. When the physician first enters the algorithm, the clinical summary box contains only a statement of the problem (Figure 1). Following

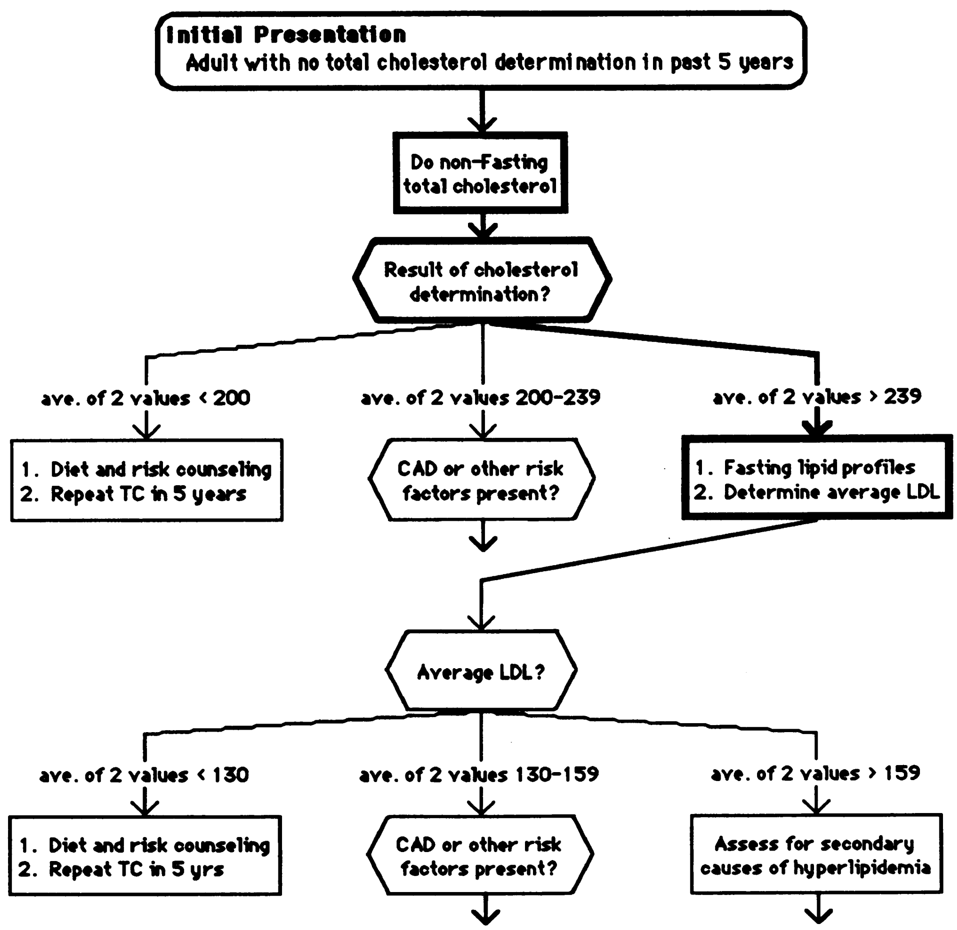

Figure 2. Computer display after the user has selected the third path (> 239) from the branch point in Figure 1. The path leading to this selection is highlighted, and the algorithm is expanded to indicate potential later steps to be taken, depending on the average LDL value.

the clinical summary box is the next action to be performed, and then a branch point which shows where the algorithm could lead after this next action.

Once the next action is performed, the user selects the appropriate path from the branch point. For example, the user would select the right path in Figure 1 to indicate that the average of two cholesterol values was greater than 239. Upon making this selection, the chosen path is highlighted, and the algorithm expands to reveal the steps leading from this path. This is shown in Figure 2.

If desired, the user could change his or her selection by choosing another path, which would then become the path which is highlighted and expanded. Positions in the algorithm from which expansion is possible are indicated by short arrows, as shown leading from the two boxes in the lower right of Figure 2.

After selecting a path, the user may leave the display expanded (as in Figure 2) or may collapse the new information provided by this selection into the clinical summary box (Figure 3). This collapsing technique provides a consistent display which is applicable to any algorithm. Regardless of the user's position in the algorithm, the display may be optionally collapsed to four parts: the clinical summary, the next action to be performed, a branch point depending on the results of this action, and potential actions leading from this branch point. The simplicity of this convention is appealing, and the size of the display remains manageable even as one moves further and further down the algorithm. Each path selection can be recorded by a single line in the clinical summary box, which itself serves to keep the user oriented to his or her logical position in the algorithm.

The clinical summary also helps to maintain user orientation when viewing the narrative discussion associated with each step in the algorithm. For example, if the user elects to examine the text supporting the second box in Figure 3, the window shown in Figure 4 will appear. In addition to discussing the rationale for this "Next Action" of the algorithm, the window includes a clinical summary which defines the context for this action. If the user reviews the discussion for a different step in the algorithm, the clinical summary included with that discussion will reflect the (different) path followed to reach that step. In general, whenever supporting text is presented, it is accompanied by its clinical context.

Maintaining orientation using an abstract view

The clinical summary gives a straightforward, detailed account of what is already known and what must be done next. It does not, however, provide a method for hiding details. Especially with a complex algorithm, an abstract view depicting only the higher level goals can help the user understand the general structure of the algorithm, and the functional relationships between its major parts.

For this, the user must be able to collapse portions of the algorithm into more abstract goal-oriented representations when detail is not needed, and to re-expand these abstract representations when a detailed view is appropriate. Each abstraction must capture the function or goal of the portion collapsed, while hiding the details of how that goal is achieved. For example, consider an algorithm for the emergency care of acute asthma. The early portions of this algorithm might include detailed suggestions for oxygen administration, inhaled and IM medications, IV fluids, and respiratory therapy. One could collapse all of these details into a single abstract step called "Administer initial treatment." When the physician enters the algorithm and needs to see the details of initial treatment, the specifics of the various therapies can be displayed. Once initial treatment is administered, the physician's interest will shift to the step "Assess response to initial treatment." At this point, the details of "Administer initial treatment" can be hidden, and those of "Assess response to initial treatment" displayed. At any time, the physician need display only those details which are of

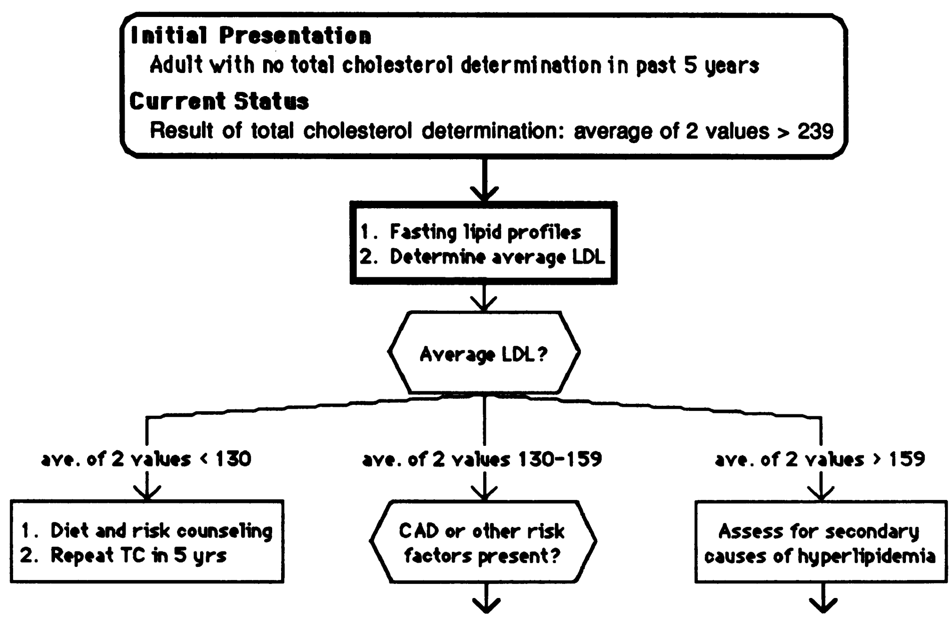

Figure 3. Display after the user has collapsed the cholesterol determination portion of Figure 2 into the clinical summary box. The next action to be performed (in this case) is a determination of average LDL. Paths leading from the "Average LDL?" branch point again indicate potential later steps to be taken, depending on the average LDL value.

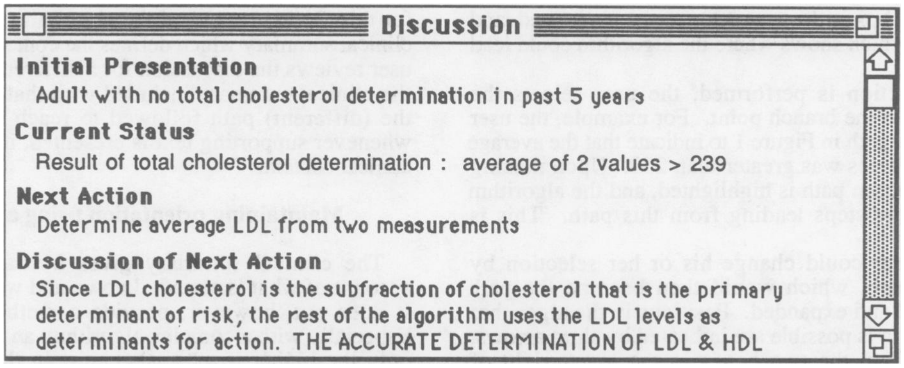

Figure 4. This figure shows part of the narrative discussion associated with the second box in Figure 3. In addition to discussing the action suggested by this box ("Determine average LDL from two measurements"), the display indicates the clinical context in which this action is recommended.

interest.

Note that the therapies abstracted under "Administer initial treatment" might themselves involve mini-algorithms (for example, to determine frequency and dosage of medications) which could be further abstracted under: "Administer initial medications," "Administer oxygen," etc. The abstract step "Administer initial treatment" subsumes the specific therapeutic modalities that constitute initial treatment: oxygen administration, medications, and fluid and respiratory therapy. In the same way, a single step called "Administer initial medications" buries the details of administering inhaled agents, epinephrine, and steroids (if needed).

The point is that multiple layers of abstraction may be useful, allowing the user to display only as much detail as is needed. Abstraction could be nested to an arbitrary level, perhaps mimicking the goal-subgoal paradigm often used in expert systems.

In addition to clarifying the general flow of clinical algorithms, abstraction techniques may allow algorithms to serve as suggested guidelines which could be followed without strict adherence to specific details. Even when disagreement exists concerning the details of a diagnostic or therapeutic strategy, there may be consensus regarding the higher level goals of the strategy. For example, two experts may disagree on the criteria for determining which acutely asthmatic patients should receive steroids, the optimal dose and preferred agent, and the duration of the steroid course. They might agree, however, that each asthmatic should be assessed after a short period of intensive therapy, with steroids administered at that time if appropriate. By representing this common ground as the abstract step "Begin steroids if indicated," both experts may accept this summary of *what* should be done, without agreeing on exactly *how* to do it. By allowing physicians to view clinical algorithms at various levels of detail, abstraction techniques might provide one method for decreasing the perceived rigidity which is frequently cited as a weakness of flow charts.

The benefits of abstraction, however, do not come without cost. One drawback is that the display becomes less consistent, since various portions of the algorithm can be collapsed or expanded independently. When there are multiple layers of abstraction, just moving through the layers may be disorienting. Thus, while abstraction techniques may be powerful, we do not yet have a clean, consistent method for exploiting this to advantage.

The clinical algorithm for advising and critiquing

To be more successful than their printed counterparts, computerized algorithms must provide functionality which the printed versions do not offer. This added functionality might be more flexibility in the display, integration with other decision support and knowledge retrieval tools, or novel methods of using the expert opinion embodied in algorithms. To illustrate the third possibility, imagine using a clinical algorithm to provide the expertise for an advising and critiquing tool.

First, the tool would need to be informed of any potentially relevant data which is already known. Ideally, much of this information would be transferred automatically from the patient's electronic medical record. With our current implementation, however, the information must be entered manually, using the interface shown in Figure 5.

This display in Figure 5 consists of three parts. The top portion provides a list of items which are potentially relevant in determining an appropriate strategy. The physician enters known values for any of these items by selecting from popup menus. In Figure 5, the physician is about to respond "no" to the question "CAD or other risk factors present?"

The bottom of the display provides additional information about the data items listed in the top of the display. In this case, the bottom portion enumerates what is meant by "other risk factors."

Once all of the known data are entered, the physician selects the "Evaluate data" button in the middle portion of the display. With this, the advising and critiquing system searches the algorithm to find a point which exactly matches the information already known. If such a point is found, the system suggests the next diagnostic or therapeutic manuever, providing a discussion of the recommended action. The display format for this recommendation is similar to Figure 4, presented earlier.

If no matching point is found, the advising/critiquing tool indicates that the entered information was obtained in a sequence different from that suggested by the algorithm. The system then identifies the piece(s) of information obtained "out of order," and explains the reasoning behind the order proposed by the algorithm. In this case, the advising/critiquing tool cannot recommend an appropriate strategy, since the information already known does not exactly match any position in the algorithm.

The advising and critiquing system is just one alternative method for accessing the expert judgment contained in algorithms. Other methods may also be helpful, since any

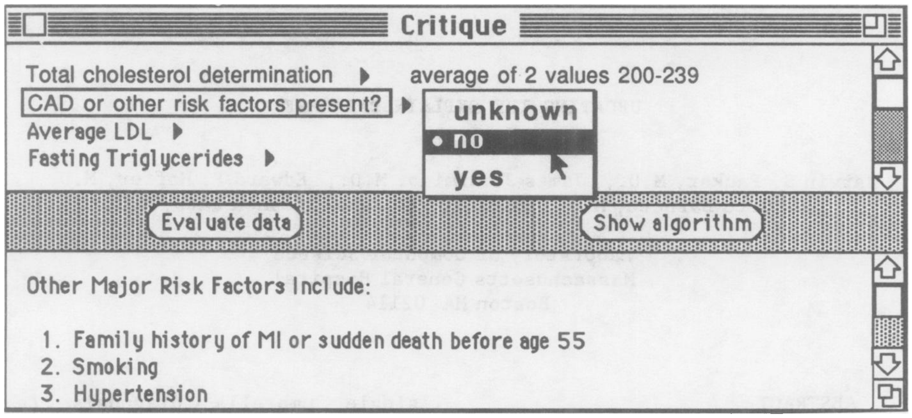

Figure 5. Current interface for data entry for our advising/critiquing tool. See text for discussion.

particular presentation may not appeal to all users or be appropriate for all situations. As one example of other methods of access, our current implementation optionally presents the branching question-and-answer logic of the algorithm in a text-only mode, without displaying boxes and arrows.

Conclusions

With computer presentation, clinical algorithms may play a valuable role in decision support and knowledge retrieval. By concisely portraying the often complex structure of diagnostic and therapeutic strategies, algorithms establish a framework for considering the relevant information. By providing links to supporting material, algorithms can guide the physician to pertinent background knowledge. By defining the specific context in which decision support is needed, algorithms might help to anticipate the kinds of support most appropriate.

While computer implementation can provide flexibility not possible using printed media, there are a number of challenges in designing an effective computer presentation of clinical algorithms. We have developed techniques for handling some of these challenges, while others remain areas of active investigation.

Clearly, data will often in practice be obtained out of order with respect to an algorithm. We are exploring an extension of the above approach which augments each algorithm with a set of rules that indicate which data items can subsume others (e.g., results of more specific tests), or function as surrogates for others, so that missing data will not preclude an otherwise valid path.

Acknowledgment

This work was partially supported by Grants LM03707 and LM04572 and Contract LM63523 from the National Library of Medicine, DHHS.

References

1. S Greenfileld *et al.*, "Efficiency and cost of primary care by nurses and physician assistants," *N Engl J Med* 298:305-309, 1978.

2. Komaroff AL, "Algorithms and the 'Art' of Medicine," *AJPH* 72:10-12, 1982.

3. Shortliffe EH, Buchanan BG, Feigenbaum EA, "Knowledge Engineering for Medical Decision Making: A review of Computer-Based Clinical Decision Aids". In: Clancey WJ and Shortliffe EH (eds), Readings in Medical Artificial Intelligence. Reading MA, Addison-Wesley Publishing Co, 1984.

4. Mesel E *et al.*, "Clinical Algorithms for Cancer Chemotherapy - Systems for Community-Based Consultant-Extenders and Oncology Centers," *Meth Inform Med* 15:168-173, 1976.

5. Wirtschafter D *et al.*, "A Consultant-Extender System for Breast Cancer Adjuvant Chemotherapy," *Ann Int Med* 90:396-401, 1979.

6. Margolis CZ, "Uses of Clinical Algorithms," *JAMA* 1983;249:627-632.

7. Eisenberg RI and Amberg JR (eds), Critical Diagnostic Pathways in Radiology: an algorithmic approach. Philadelphia, JB Lippincott Co, 1981.

8. McNeil BJ and Abrams HL (eds), Brigham and Women's Hospital Handbook of Diagnostic Imaging. Boston, Little Brown & Co, 1986.

9. Editors of Patient Care Magazine: Patient Care FlowChart Manual, ed 3. Oradell, NJ, Medical Economics Co Inc, 1982.

10. *Cardiology Emergency Decisions*. PW Communications, Inc.

11. Eisman B and Wotkyns RS, Surgical Decision Making. Philadelphia, WB Saunders Co, 1978.

12. Miller RA *et al.*, "Quick Medical Reference: A Microcomputer-Based Adaptation of INTERNIST-I." Proceedings of the Fifth Conference on Medical Informatics, Washington, D.C., October 26-30, 1986. New York, Elselvier Science Publishers, 1986.

13. Barnett GO *et al.*, "DXplain: Experience with Knowledge Acquisition and Program Evaluation." Proceedings of the Eleventh Annual Symposium on Computer Applications in Medical Care, Washington, D.C., November 1-4, 1987. New York, IEEE Computer Society Press, 1987.

14. Bergeron BP and Greenes RA, "Intelligent Visual Input: A Graphical Method for Rapid Entry of Patient-Specific data." Proceedings of the Eleventh Annual Symposium on Computer Applications in Medical Care, Washington, D.C., November 1-4, 1987. New York, IEEE Computer Society Press, 1987.